ASHESI UNIVERSITY

DESIGN AND CONSTRUCTION OF A SMART SHOPPING TROLLEY

CAPSTONE PROJECT

B.Sc. Electrical & Electronics Engineering

Andrea Adjoba Mensah

2020

ASHESI UNIVERSITY

DESIGN AND CONSTRUCTION OF A SMART SHOPPING TROLLEY

CAPSTONE PROJECT

Capstone Project submitted to the Department of Engineering, Ashesi University in partial fulfilment of the requirements for the award of Bachelor of Science degree in Electrical & Electronics Engineering.

Andrea Adjoba Mensah

2020

DECLARATION

I hereby declare that this capstone is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature: A.A.M

Candidate's Name: Andrea Adjoba Mensah

Date: 29th May,2020

I hereby declare that preparation and presentation of this capstone were supervised in accordance with the guidelines on supervision of capstone laid down by Ashesi University.

Supervisor's Signature:

………………………………………………………………………………………………

Supervisor's Name:

………………………………………………………………………………………………

Date: ………………………………………………………………………………………………

Acknowledgements

This project would not have been successful without the help of my supervisor, Mr. Francis A. Gatsi, whose academic advice, constant encouragement and support helped me undertake this project. Also, I heartily thank the entire Engineering faculty for the knowledge and skills acquired over the years.

Abstract

Shopping at a supermarket is an everyday human activity and with the increasing convenience to access supermarkets, overcrowding and the tired use of trolleys have become a problem. This project integrates many subsystems in the IoT domain to devise a solution to cut down on overcrowding and ease the burden of using shopping trolleys. A smart shopping trolley that automatically follows the user and provides opportunity for automated billing was designed in this project. The smart shopping cart is fitted with the ability to scan products to update an inventory, transmit data to a cashier point-of-sale and track where the shopper moves and directly follow. Added security to the cart ensures the cart is safe from being used by the wrong user and running into objects. The automated shopping trolley will make shopping easier and faster by providing a shopping experience that frees shoppers in supermarkets from directly handling trolleys and helps them circumvent the issue of long queues.

Table of Content

List of Figures

List of Tables

Chapter 1 : Introduction

1.1. Background

In contemporary times, shopping has become an integral part of society. In big cities and malls, traditional means of shopping has become an extreme sport in the sense that the shops are usually overcrowded, there are long queues at the billing counters, and it is hectic finding trolleys and pushing trolleys. Also, tracking of items becomes a difficult task since shopping with both hands can be inconvenient and tiring.

A supermarket typically averages 35,000 square feet [1] with about an average 23,000 users per week [2]. Supermarkets provide users with self-service to procure goods and services ranging from clothing, food, electronics to many other common needs. Supermarkets are arranged in aisles with goods categorized by section. Between aisles, shoppers traverse the supermarket with baskets or shopping trolleys to pick up goods to be purchased. Shopping trolleys provide the convenience of being mobile and able to carry huge loads. Cash counters are situated around the mall for shoppers to pay and bag their purchased items. The shopping trolley is the entry point into one's shopping experience, and depending on the quantity of goods to buy, having a shopping trolley might be a life saver. In overcrowded malls, shopping trolleys are difficult to secure, and most of them are out of shape, making shopping harder or posing dangers to customers who try to navigate around with them; the average area covered by a shopping trolley is about 4.675 square feet. To keep up with modernization and the burgeoning use of technology, a smart shopping system provides comfort of use, ease of access, and minimizes the overcrowding problem and its discomfort.

1.2. Problem Definition

Shopping centers, like big supermarkets, are overcrowded during shopping hours, especially in the holiday season. These shopping trolleys become unbearable to move around when loaded or after pushing them around for some time, that is the inconvenience of pushing or pulling the trolley and the discomfort it comes with. Furthermore, shoppers get faced with long queues when accessing the cashier, whether to request expenses to budget their shopping needs or to make final goods payment.

1.3. Solution

The automated shopping trolley, when implemented, will make shopping easier and faster by providing a shopping experience that frees shoppers in supermarkets from directly handling trolleys, providing a trolley that automatically follows the user at a maintained distance and at the same time tracks items for seamless payments. The prototype of this proposed solution will be tested in a simulated environment to test the friendliness of use, and how well it solves the problem. This smart system is of great benefit to customers who can track their shopping to stay within budget, whilst also saving time at billing counters.

1.4. Objectives

This project seeks to develop a smart shopping trolley that will improve the shopping experience of a user by reducing waiting time in queues, free a shopper from directly handling the trolley and automatically update the cashier's point of sale (POS) application with a shopper's information. The objectives are:

A. Design an automated shopping trolley that follows the shopper
B. Reduce queuing times when accessing the cashier after shopping
C. Provide a real time inventory database that updates according to shopping activity
D. Give a mechanism for shoppers to checkout their items

Chapter 2 : Literature Review

Reducing time spent in shopping malls for customers is a problem worth solving in order to free customers from having to join long queues and have a seamless smart shopping experience. The current design of the shopping trolley was introduced in the late 1940s. With so many devices getting the technological remodeling, the shopping trolley has been left out of the loop, despite its everyday use. In 2012, Chaotic Moon Lab tested out its Project Sk8, an autonomous shopping trolley controlled with Microsoft Kinect and a tablet [3]. The use of expensive technology makes this approach unrealistic, as it will cost too much to get the same number of autonomous trolleys to replace the traditional ones. Other approaches have been more experimental and took a proof-of-concept approach. However, they did not factor a lot of real-world scenarios into play like child safety, the variations of shopping load, and thefts [4][5]. These are all huge problems that many mall owners would rather have addressed to benefit them first. In places where the traditional trolleys just seem to work and owners are profit-driven, it is utterly impossible to sell these ideas to them without direct benefits.

The various types of existing smart trolleys include:

A. **RFID and IR based Smart Shopping Mart Management System.** Singh et al. propose a smart shopping trolley solution based on RFID and IR technology [6]. The system involves tagging products on shelves and using conveyor belts to deliver them to the trolleys fitted with RFID readers to allow seamless payments of goods.

B. **Automatic Shopping Trolleys using Sensors.** Deepali et al. design an automatic shopping trolley equipped with infrared (IR) sensors and an RFID system for tracking items placed in the trolley just as in [6]. The trolley follows the customer and stops when the customer stops moving. One difference Deepali et al. make from [6] is to allow the bill to be sent to the counter to be paid for.

C. Other solutions to provide a smart shopping trolley revolve around using RFID tags to scan items to generate a bill [7][8] or barcodes can be used instead of RFID tags due to their smaller size and existing adoption on products [5]. The solution provided by Viswanadha et al. introduces the option to allow customers to pay directly using mobile payment platforms [5].

Even though the solution in [6] targets the rural areas and is considered affordable, it introduces more costs and infrastructure changes that mall owners have to incorporate like the use of a conveyor belt. The sole use of RFID tags or barcode technology also makes it difficult to easily remove products already scanned, and when the products are heavy, it will be difficult to locate the tags for scanning. Allowing customers to pay themselves as suggested in [5] also has security risks where it can be impractical to know if a customer did not scan all goods, and eventually, still introduces the need for queues to check valid payments.

There are two mainstream solutions to improving time spent paying for goods and making shopping seamless and bearable. The solution from Amazon called Amazon Go gets rid of using trolleys and counters entirely and instead uses artificial intelligence-based systems to track items and take payments [9]. A shopper walks into the smart Amazon Go store after signing up on the app, and scanning the unique barcode generated. Once in the store, cameras all around the store are able to map each customer in the store. The cameras, with the use of AI, track what the customer picks off the shelf and helps customers locate items. Weight sensors on the shelves give added accuracy to determining what has been removed from a shelf. Once the shopper is done picking items, he or she walks out the store and is automatically billed. Another solution from Silicon Valley based company Focal Systems provides a shopping trolley equipped with a tablet that is able to scan the barcode of products picked off the shelf [10]. This allows customers to be automatically billed when they go to the counter to pay. Added features include in-store navigation to find products searched on the tablet, advertising of discounts and real-time stock detection.

Chapter 3 : Design Requirements

The shopping trolley system can be divided into several subparts: wireless communication, display of items in trolley, detecting shopper and planning motion path, identification of goods, microcontroller and processing, and security of the trolley. Each subpart can be implemented by different hardware components and methods:

- Wireless Communication: Wi-Fi, ZigBee, Bluetooth, Infrared
- Display: LCD, Android Tablet, Electronic Paper, Personal Phone
- Motion (Path Planning): Ultrasonic sensors, LIDAR, RADAR, Computer vision
- Goods Identification: Barcode, RFID, Computer vision
- Microcontroller: ATmega328, PIC, 8051, Renesas
- Securing and Location: GPS, Fingerprint sensor, Wireless key, Mobile App

Advantages and disadvantages of every choice in each subpart have been compared and analyzed in the concept selection process based on the system and user requirements below. The sub-functions of the system are categorized to work independently from each other as components and synergize completely to form the final trolley system. Therefore, the most direct way to select a design is to choose the optimal solution for each sub-function and combine them into the final design concept.

3.1 Requirements Specifications

This is a description of the features of the project which ensures the system functionality and the user's specifications are met. The requirements of this project is divided into two major parts: the System Requirements and the User Requirements. The system requirements lists the various technical functionalities of the project and the user requirements outlines the various expectations of the user in the development and design of this project.

3.1.1 System Requirements

- The system should go in an omni direction
- The system should be able to detect and avoid obstacles
- The system should have wireless communications for internet and data transmission
- The system should have powerful, portable battery to power all electrical systems
- The system should have a screen to display the prices of items when scanned

3.1.2 User Requirements

- The user should be able to switch between modes (automated or manual)
- The user should be able to end shopping session by scanning “end” tag
- The user should be able to view their total bill

3.2 Pugh Matrix

- **Wireless Communication**

Table 3.1 Pugh Matrix for Wireless Communication

	ZigBee	Weight	WIFI	Infrared	Bluetooth
Criteria					
Range	0	3	+3	-3	0
Power Consumption	0	3	0	-3	0
Cost	0	4	+4	+4	+4
Data Bandwidth	0	2	+2	+2	0
TOTAL			9	0	4

In selecting the wireless communication, power consumption, range and cost are the most important factors to consider. Wi-Fi provides the most efficient power usage by running the ESP32 dual Bluetooth and Wi-Fi module in deep sleep mode and only turning on at intervals when transmitting data. The Wi-Fi module has a range of about 92 meters compared with the Bluetooth and ZigBee with ranges of 10m and 90m respectively [11][12]. Based on the specifications of each module, the Wi-Fi module is the best for realizing wireless communication between the trolley and the cashier terminal and other wireless devices.

- **Display**

Table 3.2 Pugh Matrix for Display

	LCD	Weight	Android Tablet	Personal Phone	Electronic paper
Criteria					
Cost	0	5	-5	+5	+5
Battery Life	0	4	-4	0	0
User friendly	0	5	0	0	-5
Refresh Rate	0	3	+3	+3	-3
TOTAL			-6	8	-3

The electronic paper and personal phone are the least costly of the four options for displaying the shopping invoice. While the personal phone is the best option to consider since the trolley user owns it and costs nothing extra, not all trolley users will have smartphones. The use of smart phones also requires the need for a mobile app to integrate with the trolley. The electronic paper, although the next best option, lacks mainstream use due to its poor refresh rate [13]; ability to give the user a comfortable experience is a must for an improved shopping experience. In light of the benefits and disadvantages of the various display technology, the baseline LCD option is maintained and used.

- **Motion (Path Planning)**

Table 3.3 Pugh Matrix for Motion (Path Planning)

	Ultrasonic sensors	Weight	RADAR	LIDAR	Computer Vision
Criteria					
Cost	0	5	-5	-5	0
Robust to Weather and Time of Day	0	3	0	-3	-3
Performance (Speed)	0	2	+2	+2	0
Range	0	2	+2	+2	+2
TOTAL			-1	-4	-1

Autonomous motion of the trolley relies on sensors to achieve motion. By detecting objects around the trolley using the sensors, the trolley can determine the best path to follow. In this scenario, the trolley needs to identify the user and obstacles to follow the user uninterruptedly.

The LIDAR sensor is the best option for indoor detection and motion planning. However, the high cost of LIDAR makes it impractical to use. Computer vision technology with the use of cameras achieves unmatched accuracy due to the complexity in detecting colors and a wave of algorithms available for detection [14]. Computer vision is not robust to adverse conditions and relies heavily on line of sight. Also, to achieve better resolution, the cost rises significantly. RADAR is cheaper than LIDAR but not as cheap as ultrasonic sensors and cameras. RADAR can work for all kinds of ranges (short to long) and are robust to adverse conditions [15]. Since cost is a primary factor in choosing the sensor for motion, ultrasonic sensors are the best option. Multiple ultrasonic sensors coupled together can serve as accurate object detectors.

- **Goods Identification**

Table 3.4 Pugh Matrix for Goods Identification

	Barcode	Weight	RFID	Computer Vision
Criteria				
Cost	0	3	0	-3
Speed	0	2	+2	+2
Ease of Setup	0	2	-2	-2
Range	0	1	+1	+1
TOTAL			+1	-2

Computer vision with the use of cameras is fast at detecting the goods in the shopping trolley and can correctly classify any item within line of sight regardless of the range [14]. However, implementing a computer vision system requires training a machine learning object classification model, which requires data and extra computational needs aside the extra cost involved. Barcode is the mainstream technology employed in most shopping marts, yet the range of barcode is limited and requires line of sight which might prove burdensome for large goods. RFID tags can be detected easily by the reader once the tag is in range. The process of reading the tag and identification is fast and seamless since the tag can store information such as product ID, cost and other information for the invoice generation [16]. While not easy to setup compared to barcode technology, RFID is widely adopted and much easier to use than computer vision technology.

- **Microcontroller**

Table 3.5 Pugh Matrix for Microcontroller

	ATmega328	Weight	PIC	Renesas	8051
Criteria					
Memory	0	2	-2	0	-2
Performance	0	3	-3	+3	-3
Cost	0	4	+4	-4	0
Power Consumption	0	1	0	0	-1
TOTAL			-1	-1	-6

The 8051 series microcontroller from Intel is easy to set up and acquire but is slower and has less storage among the options. Renesas microcontrollers are relatively powerful and offer high performance with very low power consumption. Renesas is not as popular as the other microcontrollers, so it is more costly and lacks wide community support. Like the 8051 series, PICs are less powerful and possess lesser memory although they are the cheapest and easily acquired. The ATmega328 microcontroller, like the Renesas, has high performance and very low power consumption [17]. The ATmega328 family of microcontrollers are widespread due to their usage in Arduino development boards, and hence have large community and support for hardware components.

- **Security (Locating the trolley)**

Table 3.6 Pugh Matrix for Security (Locating the Trolley)

	GPS Tracking	Weight	Fingerprint Sensor	Wireless Key	Mobile App
Criteria					
Cost	0	3	-3	0	0
Battery Life	0	4	-4	-4	0
User friendly	0	2	0	0	-2
Size	0	1	-1	-1	+1
TOTAL			-8	-5	-1

Securing the trolley when not in use is of prime importance. The baseline option is to use GPS technology to track the position of the trolley and allow its use only within a defined fence, usually the shopping mall. Once outside the fence, the trolley wheels are locked and require special access to unlock it and return it within the fence area. Fingerprint sensors can also be used to unlock the wheels; however, the added hardware component needs more battery and space to be accommodated. A fingerprint database and registration also need to be maintained. Wireless keys or fobs can be used with each trolley that unlocks when within range of the trolley, but the use of keys means the wireless reader must be active always which will drain the battery [18][19]. A mobile application can also take the place of the above options by allowing only registered and authorized shoppers to unlock the trolleys with a mobile application. While an inventive solution, not every shopper will have a mobile phone to use the app. The best option is to use a GPS tracker to ensure the trolley is within the shopping mall vicinity.

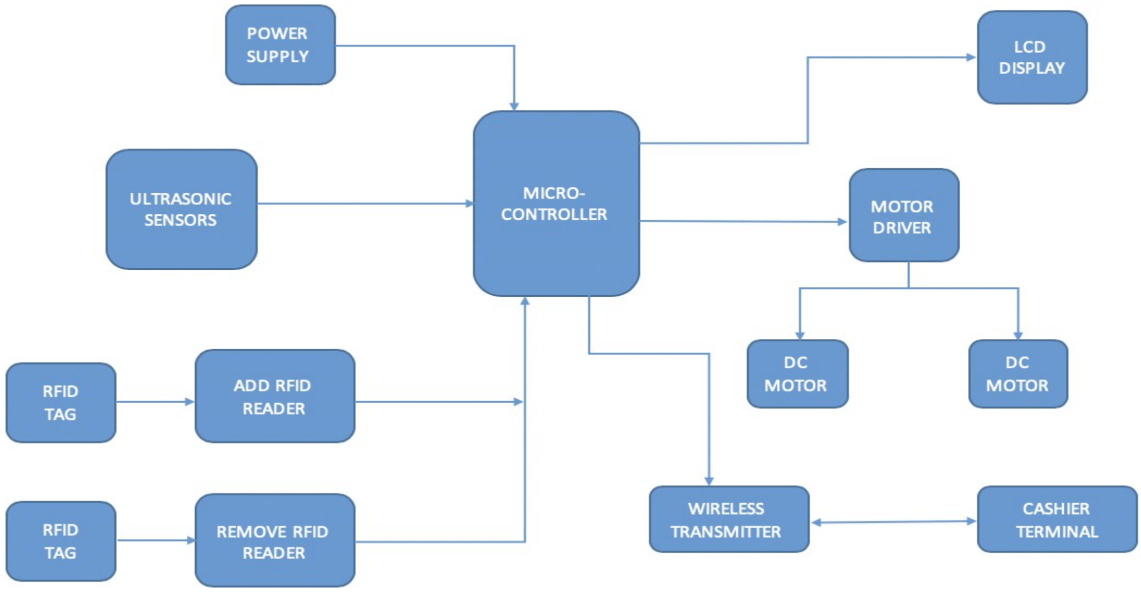


Figure 3.1 Block Diagram of design

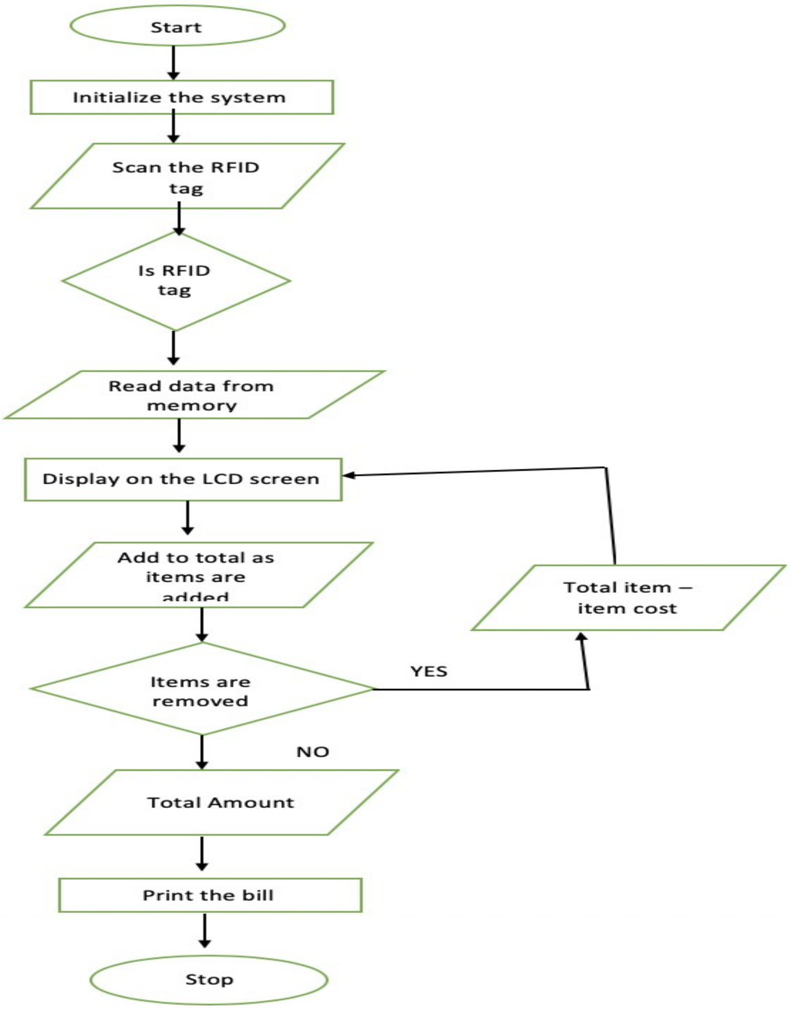


Figure 3.2 Flow Chart Diagram

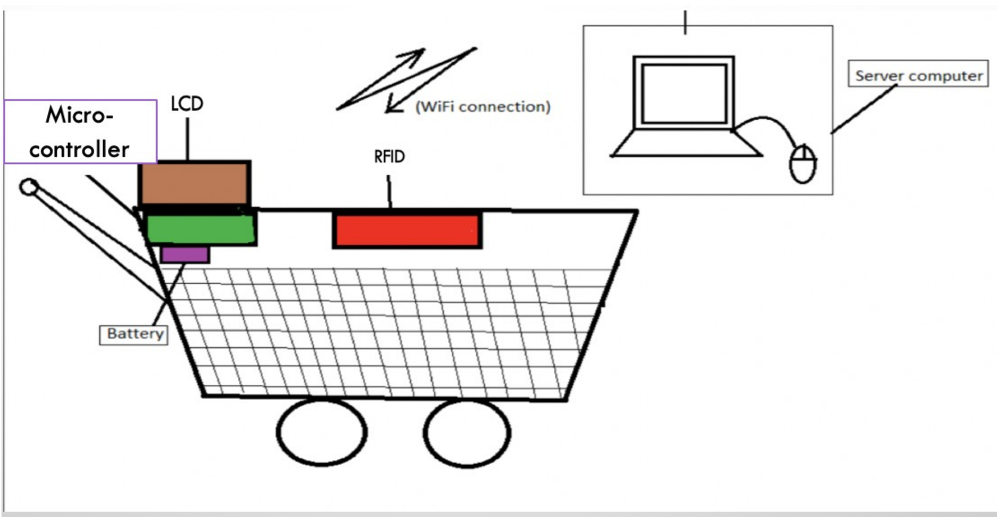


Figure 3.3 Proposed Design Sketch

Description of design

The trolley can be divided into two parts during operation: the billing and motion parts. The billing part includes scanning the items picked by the shopper, and calculating and sending the bill to the user and cashier. Every item on the store shelf is fitted with an RFID tag that the shopper can scan with the RFID reader fitted on the trolley. The cost of the item is immediately displayed, as well as the total cost needed to be paid, on the LCD display. The shopper can also remove items by rescanning the item with the push button switch held down. While the user is shopping, the trolley follows the user by determining the best path to take using the ultrasonic sensors to determine the position of the shopper, as well as obstacles. Real time sensor information is processed by the microcontroller and sent to the motor driver to power the DC motors to move the trolley forward or backward. The trolley's GPS sensor prevents the shopper from walking outside the mart with the trolley, especially when payment has not been completed. Once shopping is done, the bill is transmitted wirelessly to the cashier terminal, where the shopper can pay to the cashier without holding the line. The trolley can be directed to return to the trolley storage corner after the shopper has completed payment.

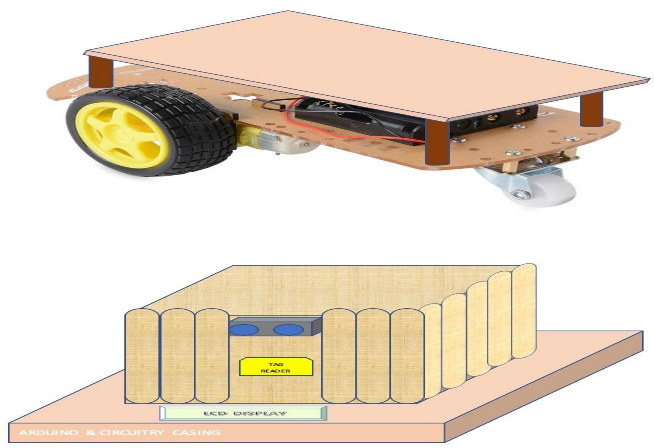


Figure 3.4 Iterated Design Sketch

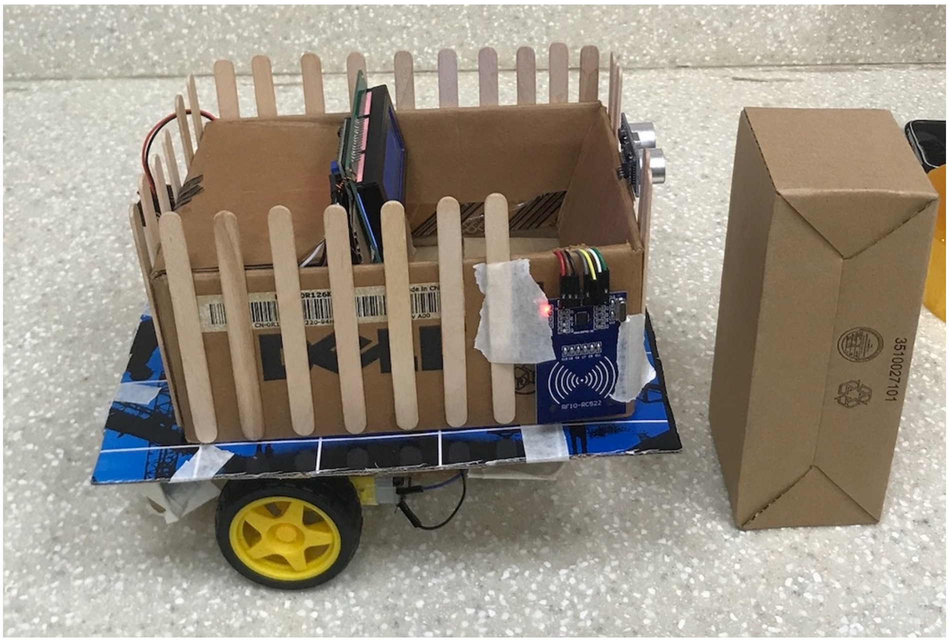


Figure 3.5 Iterated Design Prototype

Chapter 4 Methodology

4.1 Locomotion Implementation

This chapter expands on the methodology and the connection of the various sensors, actuators and processes of the project design.

4.1.1 Detection Algorithm for Ultrasonic Sensor

The algorithm for motion of the trolley requires two parts: detecting the position of a person in front of the trolley and detecting if that person is our user.

A. To detect the position of a person in front of the trolley, we use echolocation of two fixed positioned ultrasonic sensors to triangulate the location of an object. We can confirm the distance in length units of an object in front of the trolley within the range of view of the sensors.

B. To detect the person is the designated user, we use the BLE tracker tag to confirm the strength of the signal proximity.

Employing both A and B in real time will reduce errors in moving the trolley randomly for any object in its vicinity. Combining both the 2D point from A and the Received Signal Strength Indicator (RSSI) value will help us pinpoint the trolley user. Details of the algorithms employed in both A and B are below.

A. Echolocation

Setup:

1. Place ultrasonic sensors a fixed distance *s* from each other (value to be determined experimentally).
2. For one of the ultrasonic sensors *U2*, we mask out the emitter and leave out the receiver to operate it in passive mode. The other *U1* is left to operate normally in active mode.
3. Ideally, we want to place the sensors in front of the trolley as shown in fig 1.

Figure 4.1 Location of ultrasonic sensors

Procedure:

The two sensors *U1* and *U2* form a detection zone (fig 2) within which the user can be detected. An object within this zone intercepts the signal generated from the active *U1* sensor, forming a triangle that can be exploited to find the position. The triangle comprises 3 sides: the distance s between the sensors, the distance *d1* between the sensor *U1* and the target, and the distance *d2* between the target and the sensor *U2* as shown in fig 3. Using Heron's formula, trigonometry and Pythagoras theorem, the position can then be calculated as illustrated.

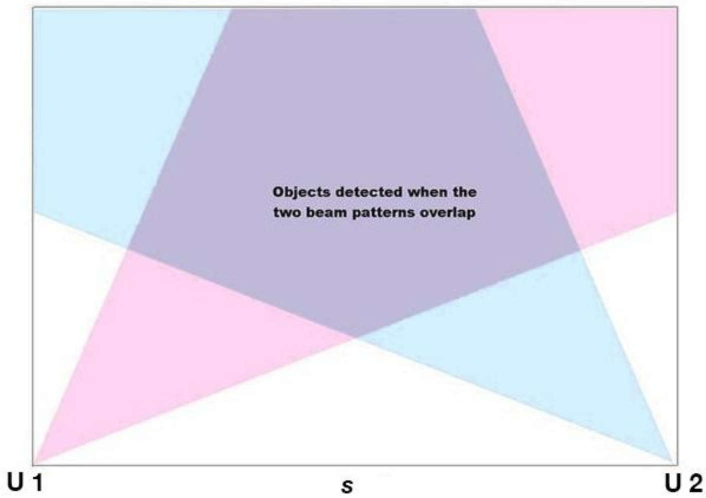


Figure 4.2 Detection zone

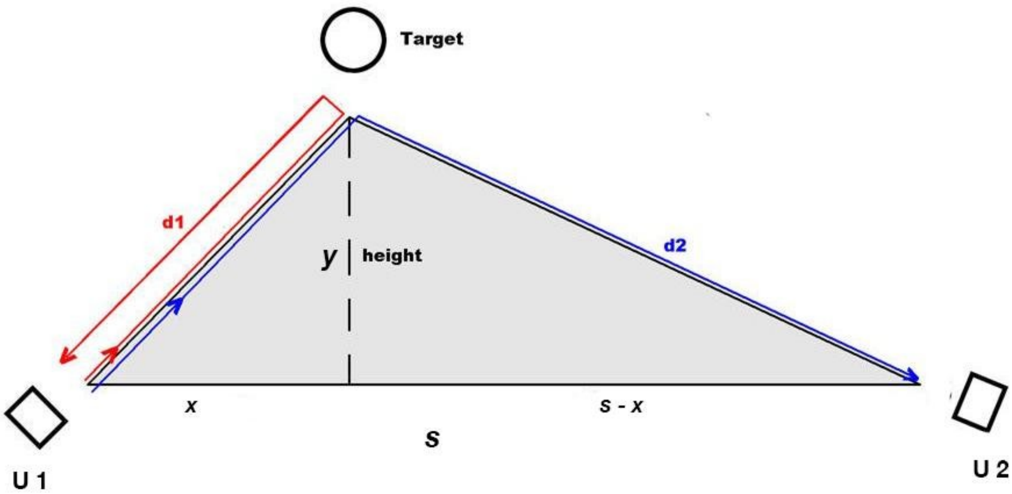


Figure 4.3 Echolocation triangle

1. For *d1*, since the time recorded is for sending the signal to the target and receiving back the signal the time is measured as:

$$time = (2 * d1)/(sound\ speed)$$

$$2 * d1(cm) = sound\ speed(\frac{cm}{us}) \times time\ (us)$$

Where sound speed = 0.034cm/us = 34/1000 cm/us = 1/29.41 cm/us

$$\boldsymbol{d1(cm) = \mathbf{time}/(29.41 * 2)}$$

2. For *d2*, since the sensor *U2* gets the echo from sensor *U1*'s signal, *d2* begins to be travelled after *d1* is travelled. This means the total time is from the signal sent to when it is received by *U2*.

$$d2 + d1\ (cm) = \text{sound speed}(\frac{\text{cm}}{\text{us}}) \times \text{time (us)}$$

$$\boldsymbol{d2(cm) = (\mathbf{time}/29.4) - d1}$$

3. Using *d1*, *d2* and the experimentally predefined *s*, we can find the area of the triangle using Heron's formula, where *c* is the semi-perimeter

$$c = (d1 + d2 + s1)/2$$

$$\boldsymbol{area = \sqrt{c(c - d1)(c - d2)(c - s)}}$$

4. Finding the Y-coordinate:

$$area\ = base * height/2$$

$$area\ = s * y/2$$

$$\boldsymbol{y = area * 2/s}$$

5. Finding the X-coordinate:

   Let *x* be the distance from *U1* to the point between *U1* and *U2* perpendicular to the target as shown in fig 3. Applying Pythagoras's theorem

$$\boldsymbol{x = \sqrt{d1^2 + y^2}}$$

B. <u>Track User</u>

Setup:

1. We store the Bluetooth MAC address of the tracker tag and associate it with the trolley in question.

2. The ESP board serves as the Bluetooth sniffer to detect the BLE tag and is mounted on the trolley.

Procedure:

1. The ESP32 starts as a client device and scans for advertised BLE devices periodically.
2. The MAC addresses of the devices are retrieved and compared against the stored MAC address.
3. If there is no match, proceed to rescan on the next iteration, otherwise, get the RSSI value of the BLE tag.
4. Compare the RSSI value to an experimentally defined value stored on the board that indicates the user is close enough.
5. Using conditional statements, indicate the user is within the vicinity if the RSSI value is within the defined constraints.

Every modern smart phone has a Bluetooth hardware capable of replicating a standard BLE device using software modifications. In this case, the Estimote App from the Apple App Store was used as a BLE device to send signals to the ESP32. We use the ESP32 reader to recognize any BLE device in range. Each BLE device has a Unique ID registered with each trolley. The device transmits its Received Signal Strength Indicator (RSSI) value in addition to the ID. The RSSI value translates to the strength of the BLE device when it is within range. Using this range and the ultrasonic sensor detection, we can localize the position of the user to trigger the trolley to move.

4.1.2 Operation of System

In general, the ultrasonic sensor behaves like a radar that finds any object in its range of detection (in this project, the intersection of the cones of both sensors to indicate the person is in between the ends of the trolley and in front). Once an object is detected, it could be a person or an object. To differentiate if it is a person, we employ the use of the tag's RSSI. The tag plays two roles through one action: it uniquely identifies that the person close by is the right owner of the trolley using the MAC address, and secondly by doing that, it clearly differentiates people from obstacles. We can extend the functionality of the ultrasonic sensor to also scan for obstacles in order not to ram into a wall or any object. But ideally, since the trolley only moves along with the person in front of it, we can be sure that it will not be moving any other time.

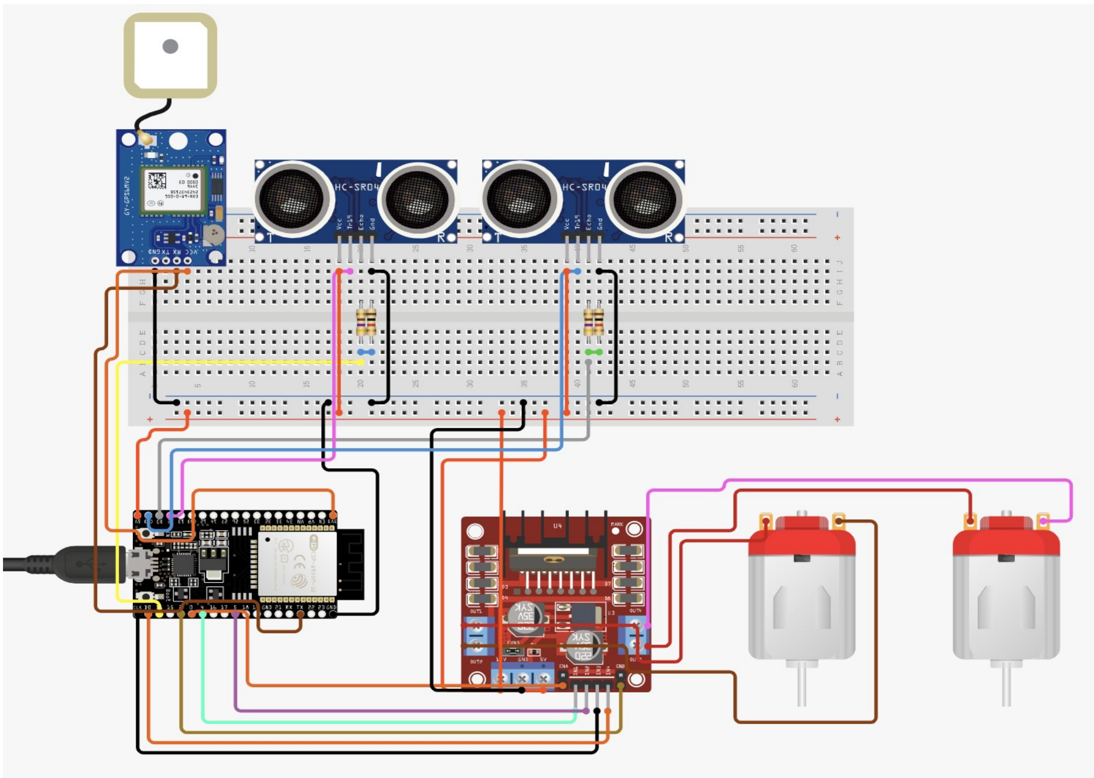


Figure 4.4 Locomotion Circuit

4.2 Communication Implementation

In the communication setup, the ESP32 model was connected to the ATmega 328 microcontroller. The ESP32 serves as the primary source of communication (master) since it houses both the Bluetooth and wireless communication hardware.

4.2.1 ESP32-ATmega328 Serial Communication

The ATmega328 microchip controls the reading and display of the prices, as well as performs the calculations of item prices needed to be sent to the POS terminal. In order to transmit the data to the cashier, the prices data needs to be sent to the ESP32 over serial connection. This is achieved by interfacing the ESP32 and ATmega328 microchip with a 5V to 3V Logic Level Convertor. The Logic Level Convertor is needed because of the voltage difference between the UART signals of the ESP32 and ATmega328. The setup of the connection is shown in the full setup diagram in Figure 4.5. By sending the data using Arduino's Serial.println function from the ATmega328, the ESP32 can read off the value into a buffer using the Serial.readBytesUntil function in the Arduino library.

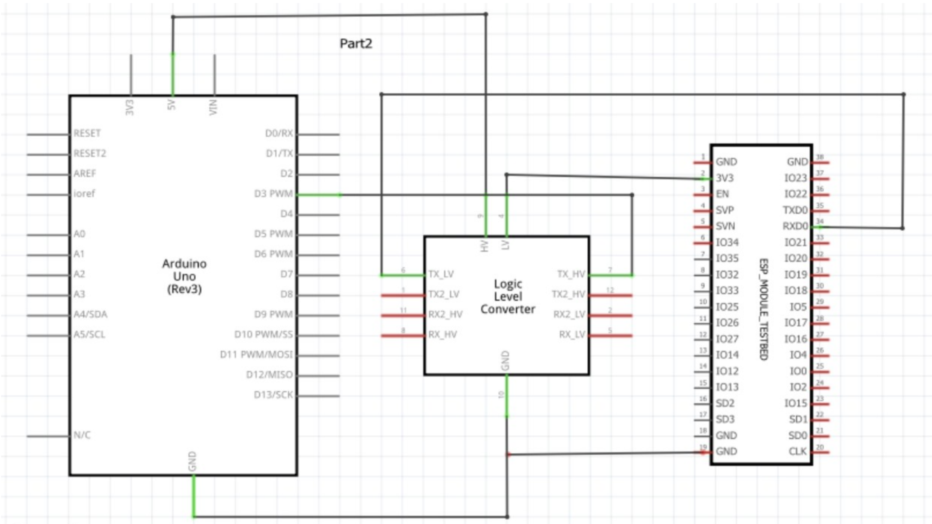


Figure 4.5 Serial Pin Connection between ATmega328 (Arduino Uno) and ESP32 chip

```
Connecting to MENSAH WIRELESS

WiFi connected
IP address:
198.168.1.201
Attempting MQTT connection...connected
{"sensorid":99,"adcValue":385}
{"sensorid":99,"adcValue":385}
{"sensorid":99,"adcValue":385}
{"sensorid":99,"adcValue":385}
{"sensorid":99,"adcValue":385}
{"sensorid":99,"adcValue":385}
{"sensorid":99,"adcValue":385}
{"sensorid":99,"adcValue":385}
{"sensorid":99,"adcValue":385}
```

Figure 4.6 Interfacing ATmega 328 and ESP32

4.2.2 ESP32-MQTT Broker Wi-Fi Communication

We employ the MQTT protocol to transmit data. The MQTT broker is the server that receives all messages from the ESP32 (client) and then routes the messages to the appropriate destination. The MQTT client is our ESP32 that connects to the MQTT broker over Wi-Fi. Bevywise MQTTRoute is used as the MQTT broker and runs locally on a PC on port 1883 and a local IP [20]. Using the Wi-Fi library, the ESP32 is connected to the wireless access point with the SSID and password. The PubSubClient library wraps the Wi-Fi instance to serve as the MQTT client library on the ESP32. For each new line received over the serial, the data is published to the broker and stored in the store database for the cashier terminal to pick up and process payment.

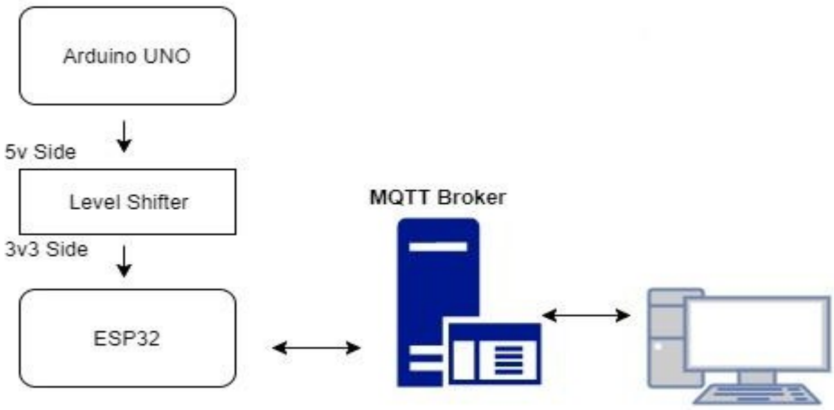


Figure 4.7 Communication between Trolley and POS terminal

4.3 Product Identification Implementation

Product identification is achieved using an RFID reader and RFID tags associated with each product. The Unique ID (UID) of each tag is stored with the price and item name in a local data table on the ATmega328 microchip. The UIDs are hashed to speed up access from the table. Once a tag is read, the microchip hashes the read UID, finds the respective item, and updates the user's cost. Using the push button while scanning the same item removes it from the list, if present in the user's trolley. To demonstrate the use of the RFID reader to scan an item, the ATmega328 and RFID reader reads a set of 4 tags associated with test goods. Successful read of the tags updates the total cost of the user and the item added, which are displaced on the attached LCD display.

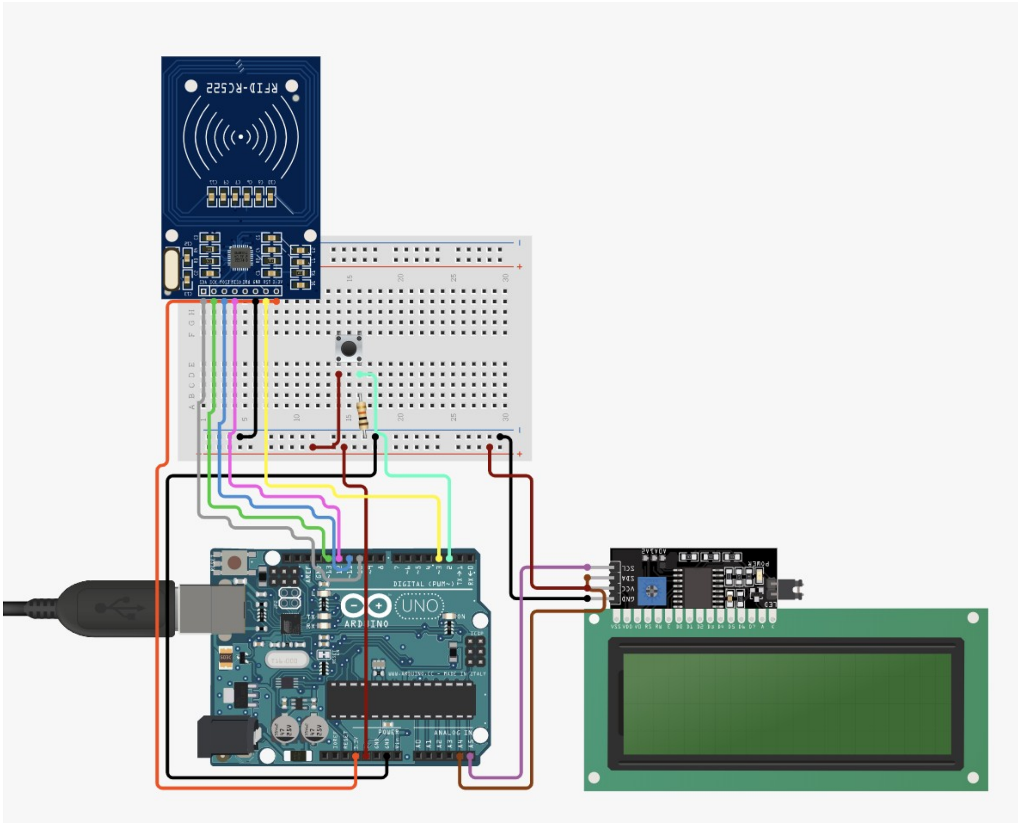


Figure 4.8 RFID Scanner Circuit

Figure 4.9 Display of scanned product

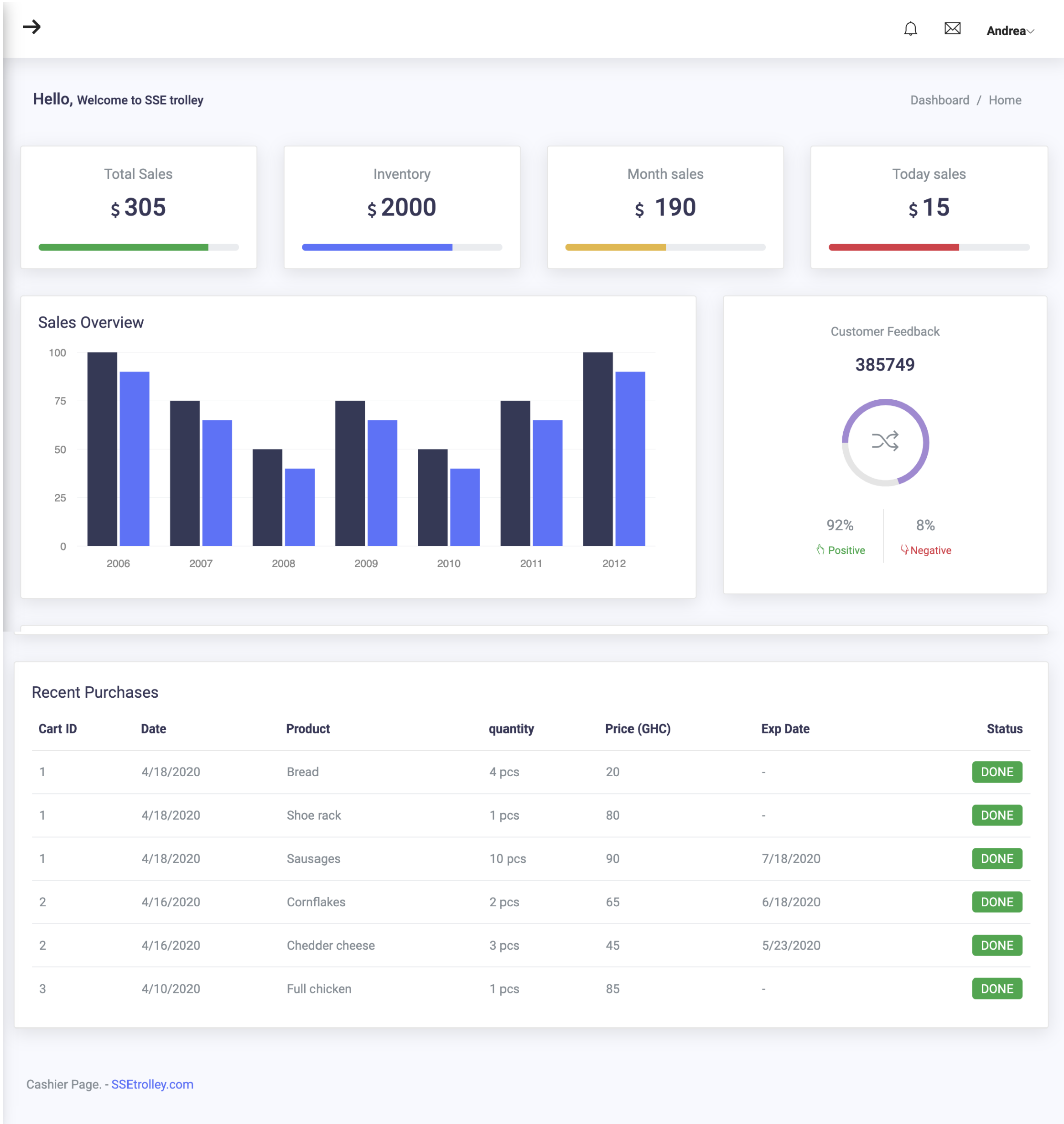
4.4 Database and POS Client Implementation

4.4.1 Database Server

A PC running the MQTT server writes data retrieved from the trolley into a NoSQL database, MongoDB. Each message published to the broker contains the ID of the user's trolley as well as a unique ID for any session if the trolley is being used by someone. A *transaction* collection holds all messages containing the items purchased as IDs published to the broker by the trolley. Another *products* collection contains the IDs of all the items in the supermarket in addition to information such as cost, name, expiry date and quantity in the store. MongoDB queries can be paired to find the data for any user and the items purchased. The query results are requested via an API and displayed to the cashier at the POS. For this project, we will implement a Node.js Server to display a simple interface dashboard that displays the cost and items of the user. The underlying code is written in standard HTML, CSS, and JS.

4.4.2 POS Client

Since the project is not focused on the design and implementation of a functioning POS client, the client use for demo purposes is sourced from Open Source POS. Open Source Point of Sale is a web-based point of sale application written in PHP using the CodeIgniter framework. It uses NoSQL as the data store at the backend and has a Bootstrap 3 based user interface [21]. Also, the *axios* JavaScript library is used to handle the GET/POST requests to the backend to display the data on the page. Only the sales function is modified and used to display data stored in the database for each user. The interface of the client is shown in Figure 4.10.



Cart ID	Date	Product	quantity	Price (GHC)	Exp Date	Status
1	4/18/2020	Bread	4 pcs	20	-	DONE
1	4/18/2020	Shoe rack	1 pcs	80	-	DONE
1	4/18/2020	Sausages	10 pcs	90	7/18/2020	DONE
2	4/16/2020	Cornflakes	2 pcs	65	6/18/2020	DONE
2	4/16/2020	Chedder cheese	3 pcs	45	5/23/2020	DONE
3	4/10/2020	Full chicken	1 pcs	85	-	DONE

Figure 4.10 POS Sales Client interface

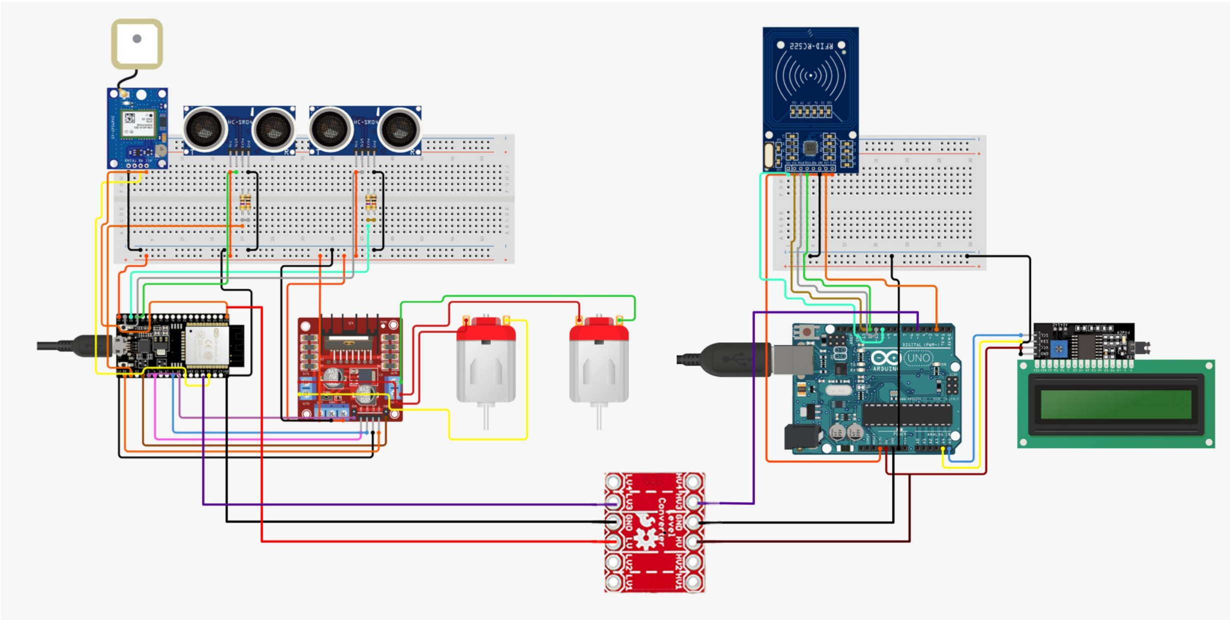


Figure 4.11 Complete setup circuit

Chapter 5: Design Testing & Results

5.1 Test Description

This chapter will focus on defined test cases that will verify our requirements for the system's design functionality. Three test cases were performed to evaluate that the design meets the user and system requirements. The test cases are:

A. Testing the user following response time to demonstrate how long it takes to start to move and stop the system. This test case also records failures for a complex user path following. For instance, if the user moves and stops 3-4 times, it records the response time as well as number of failures.
B. Testing the product identification to check the distance for successful identifications and how fast between products one can scan.
C. Testing the transfer of data using Wi-Fi to record the success and distance relation, for which a fixed dummy size data ensures that transfer data is not a bottleneck.

5.2 Test Analysis

For the first test case, we set up the trolley to follow the user in a straight path. With a preset detection distance of 10cm in front of the ultrasonic sensor, the time it takes for the trolley to start moving when the user is in range is recorded. This experiment was repeated 5 times and an average time was taken. The results obtained are shown in Table 5.1 From the values obtained, the average time was 1650ms.

Table 5.1 Response time for motion start

Trial No.	Result (ms)
1	**1624**
2	**1598**
3	**1577**
4	**1812**
5	**1639**
Average	**1650**

The recorded time taken for the trolley to come to a halt after approaching the preset safety distance of 10cm is also recorded. The results obtained are shown in Table 5.2. The average stop response time was 148.4ms, which is in the order of hundreds of milliseconds and barely noticeable, suggesting near real time response time. This is important for safety reasons as stopping the trolley from ramming into objects or people is important.

Table 5.2 Response time for motion stop

Trial No.	Result (ms)
1	130
2	128
3	135
4	183
5	166
Average	**148.4**

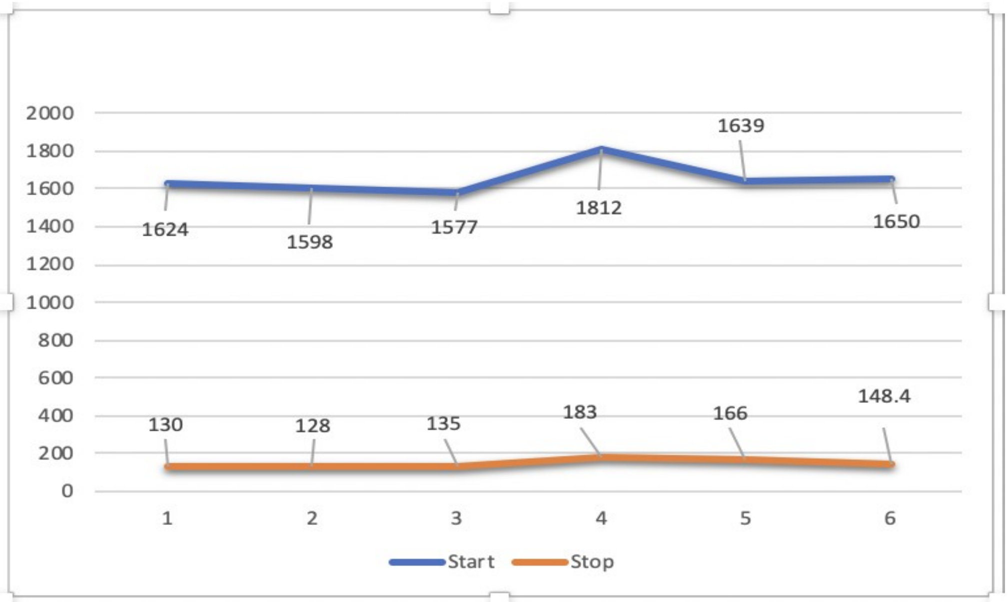


Figure 5.1 Response time graph for motion start and stop

The second test case evaluates the distance the RFID tag must be in order to successfully register the product. The tags are brought close to the reader and the distance at which the trolley updates its inventory is recorded 5 times. In another experiment, a time interval of 4s is chosen to model the fastest time a shopper may scan a product in succession. With the time interval set, tests are performed to show if the trolley updates the shopping inventory after scanning the tag in 4s succession ten (10) times. This test was repeated three (3) times with 3 different tags. The recorded values are shown in Tables 5.3 and 5.4. The average distance to record a product scan was 2.94 cm. Out of 30 instances of scanning the tag at 4s intervals, 23 successes were recorded and only 7 failures. The failures may be attributed to placing the tags on the reader wrongly and the model of the tag. The card-like tags had the most failures due to their size and difficulty in placing them close to the reader.

Table 5.3 Distance for successfully scanning product

Trial No.	Result (cm)
1	2.9
2	3.0
3	2.8

4	3.1
5	2.9
Average	**2.94**

Table 5.4 Successes/Failures (P/F) for Scanning in 4s Successions

Trial	1	2	3	4	5	6	7	8	9	10
Tag 1	P	F	P	P	F	P	F	P	P	P
Tag 2	F	P	P	P	P	P	P	P	P	P
Tag 3	F	P	P	F	P	P	P	P	P	F

In the final test case, a 1KB sized message is transmitted over Wi-Fi using the MQTT protocol to the server at the press of a button. The distance to the central location of the Wi-Fi router is varied and the success rate of the message getting to the server is recorded. At each distance, the message is sent 5 times. The results obtained are displayed in Table 5.5. The trolley was tested to a maximum distance of about 70m. At all distances, data was successfully transmitted.

Table 5.5 Success/Failure (P/F) of data transmission from trolley at varying distances (m)

Distance (m)	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
5	P	P	P	P	P
15	P	P	P	P	P
25	P	P	P	P	P
50	P	P	P	P	P
70	P	P	P	P	P

Chapter 6: Conclusion & Recommendation

This chapter explores the project limitations, shortcomings and challenges that were faced during the implementation of the project. The current system design is faced with limitations and challenges that requires further iterations of the design.

6.1 Conclusion

The automated shopping trolley system in this project is developed to enhance the shopping experience for customers by improving time spent at billing counters and making payments easier. It also frees shoppers from directly handling the trolleys by making the trolley automatically follow the user. This shopping trolley is user friendly for all types of users. Due to the variety of features with this shopping trolley, automated billing and trolley following, there is a vast difference from existing designs in the current market. Also, with this smart shopping system there will be fewer human interactions, which reduces manpower for the cashiers.

6.2 Project limitations

The design of the smart shopping system proposed to reduce overcrowded queues at billing counters and also ease the burden of shopping in terms of pushing and pulling the trolley when loaded. The limitations faced in the project are as follows:

A. Delay in acquiring some of the resources required to ensure an efficient prototype. This limitation resulted in alternative materials, which did not meet the expected results and hindered progress.

B. Given the restricted timeline already in place, the development of the coronavirus pandemic inhibited the ability to make an efficient system because of the lack of resources and tools required.

C. Restricted movement during the lockdown period due to the coronavirus pandemic made it difficult to go test the complete shopping cycle with the trolley in the field to explore any added inadequacies.
D. Inability to explore components like the Mecanum wheels to implement lateral movement of the trolley restricted it to only forward and backward motion.

6.3 Recommendations and Future work

The initial scope of this system introduces several features to make the system efficient, more robust, user friendly and secure; however, the designed system falls short of certain features that would improve the trolley's functionality. Possible improvements for this design are as follows:

A. Implementing a braking system to lock the trolley from moving when not assigned to the right user.
B. Using Mecanum wheels to add lateral movements to the trolley to follow the trolley in corners or switch lanes when obstacles are the ongoing lane [22].
C. Securing and locating the trolley within a defined fence. This feature will be implemented using a Ublox NEO-6M GPS module to locate the trolleys.
D. Adding the ability to pay for goods directly from the trolley.

The complexity of such a project required the use of multiple microcontrollers rather than just the stipulated ATMega328 chip. To make the trolley work, more pins are needed to support the added components; in this case, the ESP32 was used as both a Wi-Fi module and an added chip to supplement computations and support of other components that could not fit on the ATMega328. The communication between the two chips was done serially, which made it difficult to sync operations on both chips. A custom chip or larger chip that incorporates a wireless module as well as enough GPIO pins will be needed to model a complete trolley system.

In conclusion, the development of this smart shopping trolley system provides customers with an efficient solution that solves the inconveniences caused by pushing and pulling the trolleys as well as the long waiting queues at the counter. With a well-designed and integrated system for monitoring user location and movement, automated bill generation and tracking of items purchased, users can reduce time spent in shopping centers and also be relieved of directly handling shopping trolleys.

REFERENCES

[1] C. C. Price and D. J. Newton, "US Supermarkets: Characteristics and Services," 1986. [Online]. Available: https://naldc.nal.usda.gov/download/CAT87872507/PDF

[2] Eastwood, J. R., David B. & Gray, and Morgan D. & Brooker, "Supermarket Patronage: An Analysis of Customer Counts Among Outlets Within A Geographic Area, by Eastwood, David B.; Gray, Morgan D.; Brooker, John R.," *Journal of Food Distribution Research*, 01-Jan-1994. [Online]. Available: https://ideas.repec.org/a/ags/jlofdr/27603.html

[3] "Microsoft Kinect Games Grocery Shopping at Texas Whole Foods". *WIRED*. Feb. 27, 2012. Retrieved from https://www.wired.com/2012/02/shoppingcart/

[4] P. Deepali, C. Ashwini, and J. Nikhil, "Automatic Shopping Trolley using Sensors," *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, no. 4, 2017. Retrieved from https://www.irjet.net/archives/V4/i4/IRJET-V4I4656.pdf

[5] V. Viswanadha, P. P. Kumar, and C. S. Reddy, "Smart Shopping Cart," *2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*, Dec. 2018. Retrieved from https://ieeexplore.ieee.org/document/8821103/

[6] R. Singh, S. Verma, and M. Kriti, "RFID and IR based Smart Shopping Mart Management System," *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, 2018. Retrieved from https://ieeexplore.ieee.org/document/874

[7] A. Kumar, A. Gupta, S. Balamurugan, S. Balaji, and R. Marimuthu, "Smart Shopping Cart," *2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS)*, 2017. Retrieved from https://ieeexplore.ieee.org/document/8211723/

[8] A. Kumar, A. Gupta, S. Balamurugan, S. Balaji, and R. Marimuthu, "Smart Shopping Cart," *2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS)*, 2017. Retrieved from https://www.irjet.net/archives/V5/i5/IRJET-V5I5291.pdf

[9] M. McFarland, "I spent 53 minutes in Amazon Go and saw the future of retail," *CNN*, 03-Oct-2018. [Online]. Available: https://www.cnn.com/2018/10/03/tech/amazon-go/index.html

[10] Christynz, "Home," *InQTel*, 21-Jun-2018. [Online]. Available: https://www.iqt.org/focal-systems-selected-as-most-innovative-ai-startup-at-in-q-tel-and-silicon-valley-bank-pitch-event/. [Accessed: 20-Dec-2019].

[11] B. Mitchell, "How Far Will Your Wi-Fi Reach?," *Lifewire*, 10-Nov-2019. [Online]. Available: https://www.lifewire.com/range-of-typical-wifi-network-816564. [Accessed: 11-May-2020].

[12] "ZigBee vs Bluetooth and Bluetooth Smart - Technical Articles," *All About Circuits*. [Online]. Available: https://www.allaboutcircuits.com/technical-articles/zigbee-vs-bluetooth-and-bluetooth-smart/. [Accessed: 01-May-2020].

[13] J. Heikenfeld, P. Drzaic, J.-S. Yeo, and T. Koch, "Review Paper: A critical review of the present and future prospects for electronic paper," *Journal of the Society for Information Display*, vol. 19, no. 2, p. 129, 2011.

[14] S. Otte and S. Otte, "Understanding the Computer Vision Technology," *Innoplexus*, 30-Jan-2020. [Online]. Available: https://www.innoplexus.com/blog/understanding-the-computer-vision-technology/. [Accessed: 01-May-2020].

[15] *robotsforroboticists.com*. [Online]. Available: http://robotsforroboticists.com/lidar-vs-radar/. [Accessed: 01-May-2020].

[16] D. Henrici, A. Kabzeva, T. Fleuren, and P. Mller, "Data Storage in RFID Systems," *Radio Frequency Identification Fundamentals and Applications Bringing Research to Practice*, Jan. 2010.

[17] J. Sams, T. Agarwal, T. Agarwal, T. Agarwal, Sams, Gehad, T. Agarwal, Anderson, Harshita, T. Agarwal, T. Agarwal, Deepika, and Hi Sams Assembly, "Difference between AVR, ARM, 8051 and PIC Microcontrollers," *ElProCus*, 22-Oct-2019. [Online]. Available: https://www.elprocus.com/difference-between-avr-arm-8051-and-pic-microcontroller/. [Accessed: 01-May-2020].

[18] M.-H. Lee, H.-K. Lee, K.-H. Lim, and K.-W. Lee, "Power consumption estimation of active RFID system using simulation," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 20, no. 8, pp. 1569–1580, 2016.

[19] "RFID Key Fobs Explained," *SUMOKEY*. [Online]. Available: https://sumokey.com/pages/rfid-key-fobs-explained. [Accessed: 11-May-2020].

[20] S. Cheppali, "Arduino: Interfacing Arduino UNO with ESP32," *iCircuit*, 09-Dec-2017. [Online]. Available: https://icircuit.net/arduino-interfacing-arduino-uno-esp32/2134.

[21] Opensourcepos, "opensourcepos/opensourcepos," *GitHub*, 25-Apr-2020. [Online]. Available: https://github.com/opensourcepos/opensourcepos. [Accessed: 30-Apr-2020].

[22] Dejan, D. May, Dejan, Michael, Dejan, Dejan, M. Albano, Dejan, Jason, A. Nance, Dejan, M. Abdul, R. August, D. August, R. August, Rohith, Marcelo, Dejan, Anthony, J. Jose, Chris, D. December, D. December, H. Khan, D. February, and M. Abdul, "Arduino Mecanum Wheels